PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re application of

Roger MAITLAND, et al.

Docket No: Q102939

Appln. No.: 10/762,364

Group Art Unit: 2434

Confirmation No.: 4471

Examiner: TRAN, ELLEN C

Filed: January 23, 2004

For: METHODS AND APPARATUS FOR PARALLEL IMPLEMENTATIONS OF TABLE LOOK-UPS AND CIPHERING

RESPONSE TO NOTIFICATION OF NON-COMPLAINT APPEAL BRIEF

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In response to the Notification of Non-Compliant Appeal Brief mailed November 18, 2010, appellant submits the attached revised Summary of Claimed Subject Matter section of the brief.

Respectfully submitted,

/DJCushing/
David J. Cushing
Registration No. 28,703

SUGHRUE MION, PLLC
Telephone: (202) 293-7060
Facsimile: (202) 293-7860

WASHINGTON OFFICE
23373
CUSTOMER NUMBER

Date: May 18, 2011

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

The following is a description of the claimed subject matter with reference to each of independent claims 1, 12, 21, 35, 49-51, 55, 64, and 73-76.

The context of the invention is implementation of an encryption process which involves the use of lookup tables. Fig. 1 shows a block cipher operation wherein input data 140 is subjected to an Exclusive-OR operation at 150 to obtain ciphered data. The ciphering input to 150 comes from a block cipher 100 and is a function of a key 120 and an input 110. The present invention takes place within the cipher block 100, and is a more efficient method and apparatus for generating the output 130. In the example given in the specification, the ciphering algorithm to be implemented is a Kasumi algorithm. One of the parts of the Kasumi algorithm is an S7 function explained by the formulas of Fig. 3, where it can be seen that for each 7-bit input $X = x_6x_5x_4x_3x_2x_1x_0$ there is a 7-bit output $Y = y_6y_5y_4y_3y_2y_1y_0$ in which each bit of Y is a different logical function of the bits of X.

There are $2^7 = 128$ possible different permutations for a 7-bit X value, and therefore $2^7 = 128$ possible different values of Y. These are designated generally at 520 in Fig. 6. The 128 different values of Y are stored in a look-up table structure, but the look-up table is divided into four different tables 540, each containing 32 entries. While the look-up tables store Y values, it should be kept in mind that the labeling in Fig. 6 is not the Y values. Each different Y value will correspond to a different one of 128 possible permutations of bit values for X. Thus, in the notation of Fig. 6, "S7(0000001)" represents the S7 function of an X value of 0000001, it designates the 7-bit Y value that results from an X value of 0000001 according to the S7 function shown formulaically in Fig. 3.

Note that the values in the top look-up table 540 all correspond to X values beginning with the same most significant bits 00, the next table 540 contains entries all corresponding to X values with most significant bits 01, the third table entries all correspond to X values having most significant bits 10, and the fourth table entries all correspond to X values having most

significant bits 11. Within each table, there are 32 different values corresponding to 32 different permutations of the least significant five bits of X in combination with the particular pair of most significant bits of X. Note also that within each table 540 there is a first portion 550 containing 16 Y values corresponding to least significant X bits 00000 to 01111, and a second portion 560 containing 16 Y values corresponding to least significant X bits 10000 to 11111. (Paragraph at bottom of page 17 of the specification)

The selection process is that, for each look-up table 540, the five least significant bits of the X input are used to select one of the 32 possible Y-values at step 581 in Fig. 6, thereby resulting in four different Y-value outputs 506. (Page 18, first paragraph) In the next two selection steps 582 and 583 in Fig. 6, the value of the two most significant bits of the current X value are then used to select one of the four Y-value outputs. (Page 18, second paragraph)

Finally, while Fig. 6 illustrates the process for a given value of X, it is noted that this process takes place in parallel for plural Xi inputs, as noted at lines 4-6 of page 14 of the specification.

While the flow of the process is shown in Figs. 5 and 6 to implement the formulas in Fig. 3, the apparatus for doing so is illustrated in Fig. 19A. The look-up tables 540 in Fig. 6 are represented by memory 1810 in Fig. 19A, responding to a plurality of X inputs 1840 to produce a plurality of outputs, and the processor 1820 then uses the most significant bits to select one of the outputs for each X input, to produce outputs 1830.

Thus, in the language of claim 1, for plural inputs Xi, each input defined by a first set of bits (the five least significant bits) and a second set of bits (the two most significant bits), looking up one element from each look-up table 540 (Fig. 6) using the first set of bits to obtain a set of corresponding outputs 591-594, and then selecting a corresponding output from that set using the second set of bits. This is described in the specification in the paragraph bridging pages 13-14 of the specification.

In claim 12, and with reference to Fig. 19A, the memory is shown at 1810, processor at 1820, and the functions performed by the processor 1820 being as described above for claim 1 and as described in the specification from line 20 of page 49 to line 7 of page 50. For plural inputs Xi, each input defined by a first set of bits (the five least significant bits) and a second set of bits (the two most significant bits), looking up one element from each look-up table 540 (Fig. 6) using the first set of bits to obtain a set of corresponding outputs 591-594, and then selecting a corresponding output from that set using the second set of bits. This is described in the specification in the paragraph bridging pages 13-14 of the specification.

Claim 21 is directed to an alternative implementation shown in Figs. 11 and 20 and described in the paragraph at lines 4-30 of page 51 of the specification, where the plurality of bits making up each of N inputs are separated into subsets, Fig. 20 showing two subsets 2004 and 2006 for each of the N inputs. Each subset of input bits is used to access a look-up table, so that the N subsets 2004 yield N Group 1 outputs 2012 and the N subsets 2006 yield N Group M outputs 2014. Respective ones of the Group 1 and Group 2 outputs are then combined at 2016 to obtain N outputs 2018.

Claim 35 is directed to an apparatus for performing the method of claim 21, and this is the same apparatus as in Fig. 19A, with the memory 1810 storing the look-up table elements and the processor 1820 performing the method described in the paragraph at lines 8-18 of page 50 of the specification, where a subset of bits is used to select an element from each look-up table and then the outputs of the look-up tables are combined.

Claim 49 is directed to a computer readable medium carrying code for implementing the method of claim 1. The support in the specification for the method steps is as discussed above in the context of claim 1. For plural inputs Xi, each input defined by a first set of bits (the five least significant bits) and a second set of bits (the two most significant bits), looking up one element from each look-up table 540 (Fig. 6) using the first set of bits to obtain a set of corresponding outputs 591-594, and then selecting a corresponding output from that set using the second set of bits. This is described in the specification in the paragraph bridging pages 13-14 of the

specification. Support for the storage of code for these functions on a computer readable medium is inherent in the process being implemented by a processor as described, e.g., in the paragraph at lines 9-14 of page 14 of the specification.

Claim 50 is directed to a computer readable medium carrying code for implementing the method of claim 21. The support in the specification for the method steps is as discussed above in the context of claim 21, i.e., shown in Figs. 11 and 20 and described in the paragraph at lines 4-30 of page 51 of the specification, where the plurality of bits making up each of N inputs are separated into subsets, Fig. 20 showing two subsets 2004 and 2006 for each of the N inputs. Each subset of input bits is used to access a look-up table, so that the N subsets 2004 yield N Group 1 outputs 2012 and the N subsets 2006 yield N Group M outputs 2014. Respective ones of the Group 1 and Group 2 outputs are then combined at 2016 to obtain N outputs 2018. Support for the storage of code for these functions on a computer readable medium is inherent in the process being implemented by a processor as described, e.g., at lines 9-14 of page 16 of the specification.

Claim 51 is directed to the method shown in Fig. 20 and described at lines 4-30 of page 51 of the specification, with the bit reordering on the inputs being performed to produce M parallel sets of outputs, then for each parallel set of outputs, performing a parallel lookup table operation to generate a corresponding parallel set of outputs containing N outputs, and finally, for each of the inputs, generating a respective output by performing a bit combining operation on the outputs from the parallel look-up table operations associated with the input.

Claim 55 is directed to a ciphering method described in the specification at lines 13-23 of page 15 and lines 19-32 of page 50, and illustrated in Figs. 4 and 6, the invention being an improvement in a ciphering method illustrated in Fig. 1. The ciphering algorithm with a plurality of rounds is described in the paragraph bridging pages 10-11 and in the first full paragraph of page 11 of the specification, and shown in Figs. 2A-2D, with the plural rounds shown at 2000 in Fig. 2A (130 in Fig. 1), to generate a plurality of outputs (130 in Fig. 1) each obtained by operating on a respective input (110 in Fig. 1) using a respective key (120 in Fig. 1).

For at least one function (the S7 function in the example given in the specification), the method comprises: responsive to a plurality of first inputs (the inputs are at 1840 in Fig. 19A), the inputs are the X values each being associated with one of the respective inputs[1], for each first input and in parallel with other first inputs of the plurality of first inputs, generating an output (1830 in Fig. 1, 511 in Fig. 6, by looking up at least one look-up table (look-up tables 540 in Fig. 6) using the input, each look-up table having a plurality of elements.

Claim 64 is directed to the apparatus performing the method of claim 55, the apparatus shown in Fig. 19A, with the memory at 1810 and the processor at 1820. The functions performed by the processor and recited in claim 64 are discussed at lines 13-23 of page 15 and lines 19-32 of page 50, and shown in Figs. 4 and 6. The ciphering algorithm with a plurality of rounds is described in the paragraph bridging pages 10-11 and in the first full paragraph of page 11 of the specification, and shown in Figs. 2A-2D, with the plural rounds shown at 2000 in Fig. 2A (130 in Fig. 1), to generate a plurality of outputs (130 in Fig. 1) each obtained by operating on a respective input (110 in Fig. 1) using a respective key (120 in Fig. 1). For at least one function (the S7 function in the example given in the specification), the method comprises: responsive to a plurality of first inputs (the inputs are at 1840 in Fig. 19A), thee inputs are the X values each being associated with one of the respective inputs[2], for each first input and in parallel with other first inputs of the plurality of first inputs, generating an output (1830 in Fig. 1, 511 in Fig. 6, by looking up at least one look-up table (look-up tables 540 in Fig. 6) using the input, each look-up table having a plurality of elements.

Claim 73 is directed to a computer readable medium carrying code for implementing the method of claim 55. Support for the method steps is as discussed above in the context of claim 55, with the functions performed being discussed at lines 13-23 of page 15 and lines 19-32 of page 50, and shown in Figs. 4 and 6. The ciphering algorithm with a plurality of rounds is

---

[1] While Fig. 6 illustrates the process for a given value of X, it is noted that this process takes place in parallel for plural Xi inputs, as noted at lines 4-6 of page 14 of the specification.

[2] See footnote 1, above.

described in the paragraph bridging pages 10-11 and in the first full paragraph of page 11 of the specification, and shown in Figs. 2A-2D, with the plural rounds shown at 2000 in Fig. 2A (130 in Fig. 1), to generate a plurality of outputs (130 in Fig. 1) each obtained by operating on a respective input (110 in Fig. 1) using a respective key (120 in Fig. 1). For at least one function (the S7 function in the example given in the specification), the method comprises: responsive to a plurality of first inputs (the inputs are at 1840 in Fig. 19A), the inputs are the X values each being associated with one of the respective inputs[3], for each first input and in parallel with other first inputs of the plurality of first inputs, generating an output (1830 in Fig. 1, 511 in Fig. 6, by looking up at least one look-up table (look-up tables 540 in Fig. 6) using the input, each look-up table having a plurality of elements. Support for the storage of code for these functions on a computer readable medium is inherent in the process being implemented by a processor as described, e.g., at lines 9-14 of page 16 of the specification.

Claim 74 is directed more broadly to a method as recited in claim 55, and support for the method steps in claim 74 is found at lines 13-23 of page 15 and lines 19-32 of page 50, and in Figs. 4 and 6. Each output (130 in Fig. 1, 1830 in Fig. 19A, 511 in Fig. 6) is generated by looking up at least one look-up table (look-up tables 540 in Fig. 6) using the input, each look-up table having a plurality of elements. This is done in parallel for a plurality of inputs[4], .the inputs being the X values each being associated with one of the respective inputs.

Claim 75 is directed to an apparatus (Fig. 19A) for performing a method similar to claim 55. The memory is shown at 1810 in Fig. 19A, and the look-up tables are shown in Fig. 6. The processor is shown at 1820 in Fig. 19A, and the method steps performed are described at lines 13-23 of page 15 and lines 19-32 of page 50, and in Figs. 4 and 6. Each output (130 in Fig. 1, 1830 in Fig. 19A, 511 in Fig. 6) is generated by looking up at least one look-up table (look-up tables 540 in Fig. 6) using the input, each look-up table having a plurality of elements. This is

[3] See footnote 1, above.
[4] See footnote 1, above.

done in parallel for a plurality of inputs[5], .the inputs being the X values each being associated with one of the respective inputs.

Claim 76 is directed to a computer readable medium carrying code for practicing the method of claim 75. The look-up tables are shown in Fig. 6. A processor for executing the code is shown at 1820 in Fig. 19A, and the method steps performed are described at lines 13-23 of page 15 and lines 19-32 of page 50, and in Figs. 4 and 6. Each output (130 in Fig. 1, 1830 in Fig. 19A, 511 in Fig. 6) is generated by looking up at least one look-up table (look-up tables 540 in Fig. 6) using the input, each look-up table having a plurality of elements. This is done in parallel for a plurality of inputs[6], .the inputs being the X values each being associated with one of the respective inputs. Support for the storage of code for these functions on a computer readable medium is inherent in the process being implemented by a processor as described, e.g., at lines 9-14 of page 16 of the specification.

[5] See footnote 1, above.
[6] See footnote 1, above.